Télécom SudParis
CSC 7204
March 24th 2017
Patricia REINOSO

REPORT

A simulation of a domino game between a computer and a user was implemented. In order to accomplish this task, a graphical interface with its documentation, and part of the skeleton of the implementation was provided.

The initial requirements were also provided. These are the following:

“The domino set contains **28** dominoes.

A round begins by distributing 6 dominos to the 2 players.

The one who starts is the one who has a double six. If no one has a double six, the double five is looked for, then the double four etc. In the absence of a double, the human user will start.

When it is his/her/its turn to play:

- the player puts down a domino matching with one of the extremities of the dominos that are already on the "table".

- if the player does not have any matching domino, he/she/it takes from the stock a domino until he/she/it gets a valid one or that the stock is empty. In that latter case, the player passes and it is the other player's turn.

A round stops when one of the players does not have any domino left or that no player can either puts a domino down or take one from the stock.

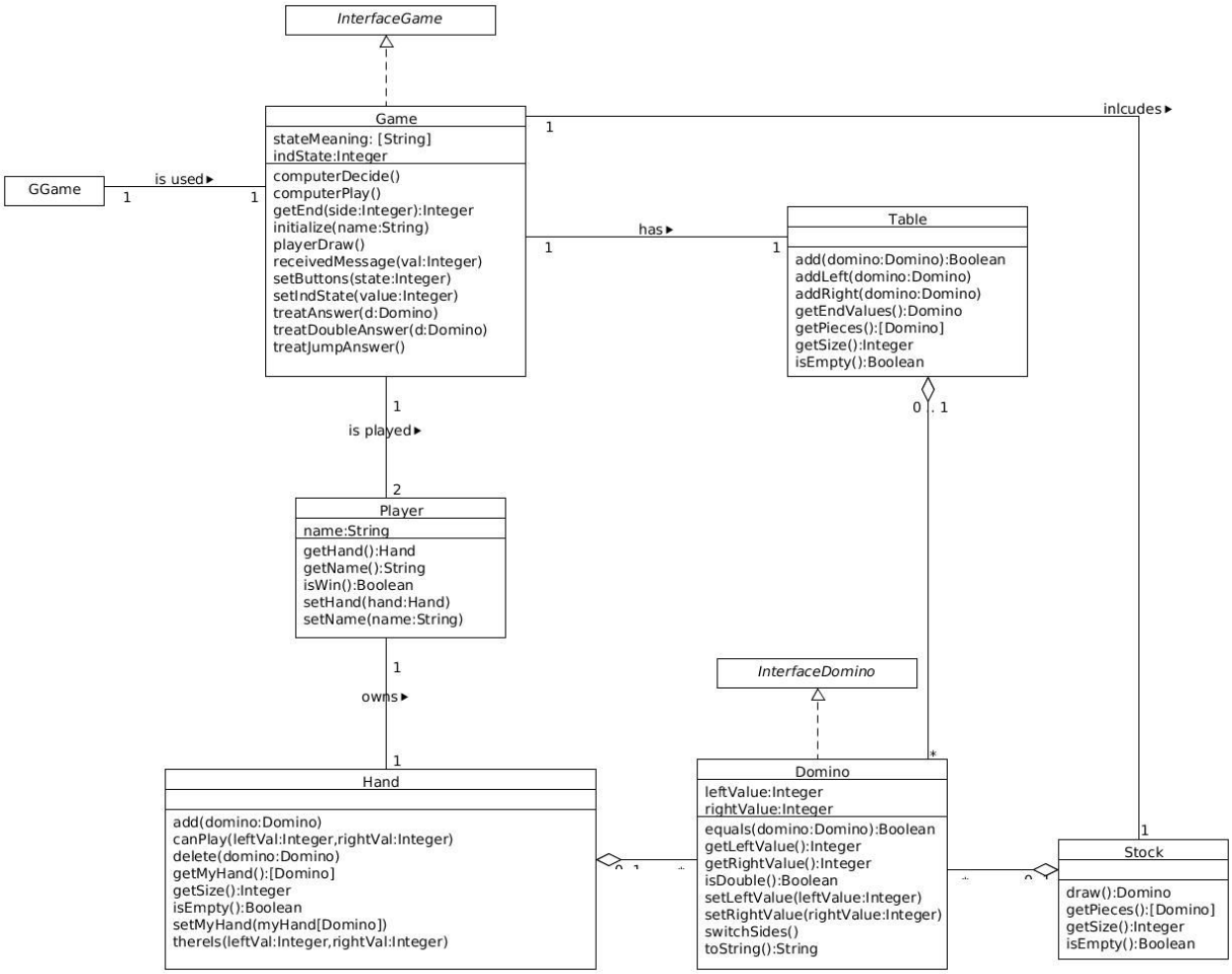
The dominos are the following:

(6,6),(6,5),(6,4),(6,3),(6,2),(6,1),(6,0),(5,5),(5,4),(5,3),(5,2),(5,1),(5,0), (4,4),(4,3),(4,2),(4,1),(4,0), (3,3),(3,2),(3,1),(3,0), (2,2),(2,1),(2,0), (1,1),(1,0), (0,0).”

Class diagram

According to these instructions of the game, and the documentation provided, a class diagram was created. Nevertheless, during the implementation of the game, some methods were required and therefore, created. And other were not used in any case, therefore, they were deleted.

The final class diagram that represents the design of the software and the relationship among the classes, is the following:



The classes: Player, Hand, Table, Domino and Stock, were created, whereas the classes Interface Domino and GGame were provided, and the classed Game was completed (only the skeleton was provided).

Game aspects considered and changes on the code.

The implementation of the game was executed based on the model game provided. Further, the following situations on the game were considered:

1. The player is allowed to jump only in 2 situations:
   - When we are searching for the first piece to put on the board (a double) and the player does not possed it.
   - When the stock is empty and the player does not have a domino piece that matches any of the ends of the board.

If the player tries to jump even if he or she possesses an appropriate domino piece to play is considered as cheating. Therefore, both situations conditions are verified when clicking on the Jump button.

2. There is no limitation on the draw Button. The player is able to draw a domino pieces, even if he or she possesses an appropriate domino piece to play. The only constraint is when the stock is empty.

3. It is not possible to select on which end-side of the table the domino piece is going to be placed (if both ends are possible). The priority given to the left value of the domino piece. This value is tried to be matched to the end-values of the table. If it does not match, then the right value is verified.

4. The numbers of the states of the game were changed in order to be more organized. The states are the following:

State	Context
0 - 6	The first domino piece has not been played. We are searching for a double domino piece.
7	The player plays.
8	The player plays and the stock is empty.
9	The computer plays.
10	It is the computer's turn, but it draws from the stock.
11	The computer is blocked (the stock is empty).
12	The computer plays with empty stock.
13	The player plays and the computer is block.

Implementation state

It is possible to play the game, and the basic functionalities of the game were implemented, nevertheless, the following parts were not concluded:

1. The button to start a new game does not work.
2. According to the documentation of the graphical interface and the model game provided, there is a difference between clicking on a domino piece using the left or the right button. Though, this was not implemented, it is only possible to select a piece using the left button.

Limitations

At the beginning, the integration of the graphical interface and the classes created was very difficult to accomplished, specially because it was the first time using a graphical interface in java. Regardless, reading the documentation was very helpful, and the skeleton provided few hints that were used as a guidance.

In addition, only validation tests were executed and, given the fact of the randomness of the game (the distribution of the pieces) it was very difficult to test some specific cases and it was required several attempts or doing modifications on the stock in order to reach the desired situations.

Conclusions

The creation of a class diagram is very helpful on the object-oriented programming paradigm. Even if the transformation on an specific programming language is not exactly, performing a mapping is not a demanding task.

Nevertheless, there on this cases, there are many situations that are very similar (only few details changed) but, as a result of the design of the software and the skeleton provided, it was necessary to repeat some parts of the code several times.

The implementation of the game was done basic on the previous experience playing the game and on the instructions provided.